

TEST CASE DEVELOPMENT				
Backend C# APIS				
API TESTING FOR USER AUTHENTICATION				
Test ID	Endpoint	Description	Request Body	Expected Result
API01	USER /api/User/login	Login - Valid	{ "username": "testuser", "password": "password"}	200 expected
API02	USER /api/User/login	Login - Invalid Password	{ "username": "testuser", "password": "wrongpass"}	401 expected
API03	USER /api/User/login	Login - Invalid Username	{ "username": "wronguser", "password": "password"}	401 expected
API04	USER /api/User/login	Login - Missing Fields		400 expected
API TESTING FOR PRODUCT MANAGEMENT				
Test ID	Endpoint	Description	Request Body	Expected Result
API05	PRODUCT /api/Product/1	Get Product by ID - Before Create		200 expected
API06	PRODUCT /api/Product	Create Product - Valid ID 1	{ "id": 1, "name": "New Product", "price": 9.99}	201 expected
API07	PRODUCT /api/Product	Create Product - Duplicate ID 1	{ "id": 1, "name": "New Product", "price": 9.99}	201 expected
API08	PRODUCT /api/Product	Create Product - Invalid Negative ID	{ "id": -1, "name": "Negative ID product", "price": 9.99}	201 expected
API09	PRODUCT /api/Product	Create Product - Invalid 0 ID	{ "id": 0, "name": "0 ID product", "price": 9.99}	201 expected
API10	PRODUCT /api/Product	Create Product - Invalid price as string	{ "id": 3, "name": "String priced product", "price": "precio de mil"}	400 expected
API11	PRODUCT /api/Product	Get All Products		200 expected
API12	PRODUCT /api/Product/1	Get Product by ID - Valid		200 expected
API13	PRODUCT /api/Product/9999	Get Product by ID - Invalid		404 expected
API14	PRODUCT /api/Product	Create Product - Missing Fields	{ "price": 9.99}	201 expected
API15	PRODUCT /api/Product	Create Product - Missing all fields		400 expected
API16	PRODUCT /api/Product/1	Update Product - Valid	{ "id": 1, "name": "Updated Product", "price": 19.99}	204 expected
API17	PRODUCT /api/Product/1	Update Product - Invalid Update ID	{ "id": 100}	204 expected
API18	PRODUCT /api/Product/1	Update Product - Update price to string	{ "id": 1, "name": "Updated Product", "price": "String Price"}	400 expected
API19	PRODUCT /api/Product/9999	Update Product - Invalid ID	{ "id": 99999, "name": "Updated Product", "price": 19.99}	404 expected
API20	PRODUCT /api/Product/1	Delete Product - Valid		204 expected
API21	PRODUCT /api/Product/9999	Delete Product - Invalid ID		404 expected
API TESTING FOR ORDER MANAGEMENT				
Test ID	Endpoint	Description	Request Body	Expected Result
API22	ORDER /api/Order/1	Get Order by ID - Before Create		200 expected
API23	ORDER /api/Order	Create Order - Valid ID 1	{ "id": 1, "productName": "Test Product", "quantity": 2, "status": "Pending"}	201 expected
API24	ORDER /api/Order	Create Order - Duplicate ID 1	{ "id": 1, "productName": "Test Product", "quantity": 2, "status": "Pending"}	201 expected
API25	ORDER /api/Order	Create Order - Invalid Negative ID	{ "id": -1, "productName": "Test Product", "quantity": 2, "status": "Pending"}	201 expected
API26	ORDER /api/Order	Create Order - Invalid ID 0	{ "id": 0, "productName": "Test Product", "quantity": 2, "status": "Pending"}	201 expected
API27	ORDER /api/Order	Create Order - Invalid Quantity as string	{ "id": 1, "productName": "Test Product", "quantity": "two", "status": "Pending"}	400 expected
API28	ORDER /api/Order	Get All Orders		200 expected
API29	ORDER /api/Order/1	Get Order by ID - Valid		200 expected
API30	ORDER /api/Order/9999999	Get Order by ID - Invalid		404 expected
API31	ORDER /api/Order	Create Order - Missing Fields	{ "quantity": 2}	201 expected

API32	ORDER /api/Order	Create Order - Missing all Fields		400 expected
API33	ORDER /api/Order/1	Update Order - Valid	{ "id": 1, "productName": "Updated", "quantity": 5, "status": "Shipped"}	204 expected
API34	ORDER /api/Order/1	Update Order - Invalid update ID	{ "id": 101}	204 expected
API35	ORDER /api/Order/1	Update Order - Invalid quantity to string	{ "id": 99999, "productName": "Updated", "quantity": "Five", "status": "Shipped"}	400 expected
API36	ORDER /api/Order/99999	Update Order - Invalid ID	{ "id": 99999, "productName": "Updated", "quantity": 5, "status": "Shipped"}	404 expected
API37	ORDER /api/Order/1	Delete Order - Valid		204 expected
API38	ORDER /api/Order/99999	Delete Order - Invalid ID		404 expected

Frontend

LOGIN TESTING

Test ID	Script File	Test Description	Steps	Expected results
LOGIN-01	login.cy.js	Logs in successfully with valid credentials	1- Go to login Page 2- Fill login form with correct user info 3- press login button	Successfull login and a token
LOGIN-02	login.cy.js	Fails login with wrong password	1- Go to login Page 2- Fill login form with wrong password 3- press login button	Fail to login
LOGIN-03	login.cy.js	Fails login with wrong username	1- Go to login Page 2- Fill login form with wrong user name 3- press login button	Fail to login
LOGIN-04	login.cy.js	Fails login with both fields incorrect	1- Go to login Page 2- Fill login form with both fields wrong 3- press login button	Fail to login
LOGIN-05	login.cy.js	Fails login with empty password	1- Go to login Page 2- Fill only username text input 3- press login button	Fail to login
LOGIN-06	login.cy.js	Fails login with empty username	1- Go to login Page 2- Fill only password text input 3- press login button	Fail to login
LOGIN-07	login.cy.js	Fails login with both fields empty	1- Go to login Page 2- press login button	Fail to login

ORDER LIST TESTING

Test ID	Script File	Test Description	Steps	Expected result
ORDER-LIST-01	order-list.cy.js	Display empty list of orders	1-Go to orders page	Renders a list of logged orders
ORDER-LIST-02	order-list.cy.js	Creates a new order	1- Go to orders page 2- Sent a POST REQUEST 3- Reload the page 4- Check for the existence of the order	New order displayed
ORDER-LIST-03	order-list.cy.js	Updates an order	1- Go to orders page 2-Sent a PUT REQUEST 3-Reload the page 4- Check for updated information	Order updated
ORDER-LIST-04	order-list.cy.js	Deletes an order	1- Go to orders page 2-Sent a DELETE REQUEST 3-Reload the page 4- Check for the order not to be rendered	Order deleted

ORDER-LIST-05	order-list.cy.js	Bulk inserts and deletes orders	1- Go to orders page 2-Sent various POST REQUEST 3-Reload the page on each iteration 4- Check for the last created order 5- Send 2 DELETE REQUEST 6- Reload on each iteration 7- Check for the orders to not be rendered	Orders created deleted and 4 orders remains
PRODUCT LIST TESTING				
Test ID	Script File	Test Description	Steps	Expected result
PRODUCT-LIST-01	product-list.cy.js	Fetches all products empty	1- go to products page 2- Ensure no item is rendered	Empry array
PRODUCT-LIST-02	product-list.cy.js	Creates a new product	1- Go to Products page 2- Sent a POST REQUEST 3- Reload the page 4- Check for the existence of the Product	Product created
PRODUCT-LIST-03	product-list.cy.js	Fetches all products	1- Go to Products page 2- Ensure the item is rendered	Render a list of products
PRODUCT-LIST-04	product-list.cy.js	Fetches a product by ID	1- Go to Products page 2- Sent a POST REQUEST with and ID 3- Check response	Product payload
PRODUCT-LIST-05	product-list.cy.js	Updates a product	1- Go to Products page 2- Sent a PUT REQUEST 3- Reload the page 4- Check for the existence of the updated product	Updated product
PRODUCT-LIST-06	product-list.cy.js	Deletes a single product	1- Go to Products page 2- Sent a DELETE REQUEST 3- Reload the page 4- Check for the existence of theProduct	Deleted product
PRODUCT-LIST-07	product-list.cy.js	Inserts multiple products	1- Go to products page 2-Sent various POST REQUEST 3-Reload the page on each iteration 4- Check for each created Product	Multiple products created
PRODUCT-LIST-08	product-list.cy.js	Deletes multiple products	1- Send 2 DELETE REQUEST 2- Reload on each iteration 3- Check for the products to not be rendered	2 products deleted, 4 remain
RESPONSIVE CAPABILITIES TESTING				
Test ID	Script File	Test Description	steps	Expected result
RESPONSIVE-01	responsive.cy.js	Displays nav and page content correctly on all rout	1- Change the viewport 2-check for the existance of the pages	all information rendered in all viewports
DASHBOAR PAGE TESTING				
Test ID	Script File	Test Description	Steps	Expected result
DASHBOARD-01	dashboard.cy.js	Should load dashboard	1- Go to dashboard page	Dashboard page should render